NAVIGATION OF AUTONOMOUS ROBOTIC VEHICLES USING FUZZY LOGIC

Nikos C. Tsourveloudis

Intelligent Systems and Robotics Laboratory
Department of Production Engineering and Management
Technical University of Crete
73100 Chania, Crete, GREECE
e-mail: nikost@dpem.tuc.gr

Abstract: A general fuzzy logic control framework along with its application specific modifications is presented to support, evaluate and justify the proposed perspective to unmanned vehicle autonomous navigation. The paper discusses successful applications of collision free motion control of ground, aerial and underwater unmanned vehicles navigation. The common characteristic in all applications regardless of the type of vehicle is the navigation architecture used. Experimental and simulation results are included to validate and support the implemented techniques and approaches. A comparison of classical and soft computing based controllers, designed to control an underwater vehicle provides additional evidence of the usefulness and applicability of fuzzy logic as a viable alternative to using analytic approaches, and as a modeling tool that deals with real life ill-defined problems.

Keywords: Autonomous navigation, unmanned vehicles, fuzzy logic.

1. INTRODUCTION

This paper is the outgrowth of on-going as well as published research performed by the author and his colleagues in the area of autonomous and collision-free navigation of unmanned robotic vehicles operating on the ground (indoors: Piperidis *et al*., 2007; Doitsidis *et al*., 2002; Tsourveloudis *et al*., 2001), in the air (Doitsidis *et al*., 2004; Nikolos *et al*., 2003; Vitzilaios and Tsourveloudis, 2007) and in underwater environments (Kanakakis and Tsourveloudis, 2007; Kanakakis *et al*., 2004).

The objective of this paper is to present implementations of fuzzy logic in the area of autonomous vehicles navigation made by the *Intelligent Systems and Robotics Laboratory* of the Technical University of Crete. The paper discusses successful applications of collision free motion control of *ground*, *aerial* and *underwater* unmanned vehicles navigation. The common characteristic in all applications that will be presented is that regardless of the type of the vehicle used, the navigation architecture remains the same. Therefore, another objective of this paper is to register and justify the layered fuzzy logic based control architecture, as applicable to any unmanned vehicle with minor modifications (Tsourveloudis *et al*., 2005). Further, focusing on highly coupled, unstable and nonlinear systems (such as underwater vehicles) that are difficult to control, a secondary objective of the paper is to justify - through comparisons between classical and fuzzy logic controller designs - that fuzzy logic is a viable solution for the real-time navigation of such systems.

The wide applicability of fuzzy logic in autonomous navigation (Driankov and Saffiotti, 2001) is mainly based on suitable knowledge representation of inherently vague notions achieved through fuzzy IF-

THEN rules. These rules typically contain linguistic information, which describes the problem at hand very simple and fast. In the majority of fuzzy logic application in navigation, a mathematical model of the dynamics of the vehicle is not needed in the design process of the motion controller. Only the problem-specific heuristic control knowledge is needed for the inference engine design. From a more practical point of view, fuzzy logic is the most appropriate modeling tool for representing imprecision and uncertainty of the sensor readings. Another reason that explains the popularity of fuzzy logic in autonomous navigation is the low computation time of the hardware implementations of fuzzy controllers which favors real-time applications.

The proposed generic concept of fuzzy navigation architecture is discussed in the next section. Section 3 presents the implementation of the proposed generic architecture for ground, aerial and underwater robots. Experimental results are presented for the case of an underwater robotic vehicle. The paper concludes with suggestions for future research.

2. A COMMON CONTROL ARCHITECTURE FOR NAVIGATION

Fuzzy inference approaches tend to de-emphasize goal-directed navigation and focus more upon handling reactive and reflexive cases. In dynamic environments, fuzzy navigation either follows a classical paradigm or a behavior-based paradigm. Fuzzy navigation schemes, which follow the classical paradigm, have one set of control rules that includes all situations that may arise. All rules operate at all times to generate the control law. Behavior based fuzzy navigation acknowledges that there are different types of behaviors which the autonomous vehicle must exhibit in different situations. Each behavior is given a set of rules and an inference engine is used to determine which behavior (or combination of behaviors) needs to be invoked in the current situation. In both paradigms, the “reaction” is given by a set of rules, which describe the navigation priorities.

The results of the fuzzy inference controllers generally do not tend towards optimal paths. However, surprise obstacles and rapidly moving obstacles are handled with more certainty compared to methodologies in which certain performance criteria should be optimized (Tsourveloudis *et al*., 2001). Regardless of the final navigation goal or the type of vehicles, some kind of sensor data management is needed. Sensor readings provide information about the environment and the vehicle itself. These readings are almost at all times erratic, incomplete or conflicting and should be further processed in order to provide meaningful information. This information is essential for the motion commands of the vehicle. The overall architecture of the proposed navigation schema is shown in Fig. 1.

The *sensor fusion* block of Fig. 1 represents a fuzzy controller which takes as input the data provided by the various sensors and delivers information for eventual obstacles in respect to vehicle’s position and orientation. The interpreted obstacle information forms a collision possibility, which is send to the *motion control* module. The collision possibility together with position and/or orientation error are inputs of the motion controller, which is responsible for the output commands to the driving devices.

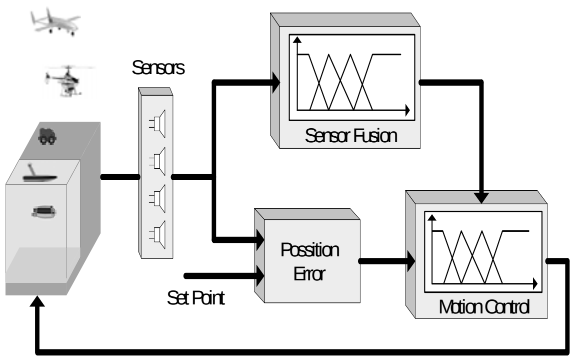

Fig. 1. Architecture of a general navigation scheme.

The architecture presented in Fig. 1 has been successfully applied to various unmanned vehicles as it is described in the following sections. In all these applications the basic idea of the layered fuzzy control is utilized in respect to the control demands of each robotic vehicle.

3. APPLICATIONS OF THE NAVIGATION ARCHITECTURE

In robotics, autonomy is mainly associated with navigation issues. Autonomous navigation of unmanned vehicles in unstructured environments is a multidiscipline and attractive challenge for researchers. From a conceptual point of view, autonomous navigation of robotic vehicles may be achieved via continuous interaction between *perception*, *intelligence* and *action.* The presentation that follows describes state-of-the-art applications of fuzzy logic that follow the architecture presented in the previous section.

3.1 Unmanned Ground Vehicles.

The navigation scheme of Fig. 1 was initially implemented on a *Nomad 200* mobile robot platform (Tsourveloudis *et al.*, 2001), for indoor navigation, and later on an ATRV skid steering mobile robot manufactured by *iRobot* (Doitsidis *et al.*, 2002). The same techniques are recently implemented on the *ALE II* and *HELOTS*, two mobile robots designed and developed by the *Intelligent Systems and Robotics Laboratory* of the Technical University of Crete (Piperidis *et al*., 2007). In all above mentioned implementations, the collision possibilities are calculated using fuzzy rules of the type:

IF d_i *is* $LD^{(k)}$ AND d_{i+1} *is* $LD^{(k)}$ THEN c_j is $LC^{(k)}$,

where k is the rule number, d_i represents sensors group i minimum readings, $LD^{(k)}$ is the linguistic variable of the term set D ={*near, medium_distance, away*}, c_j is the collision direction and $LC^{(k)}$ the variable with term set C={*not_possible, possible, high_possibility*}. The overall output of the sensor fusion block is calculated by the composition between the fuzzified sonar readings and the navigation priorities as described by the fuzzy rules.

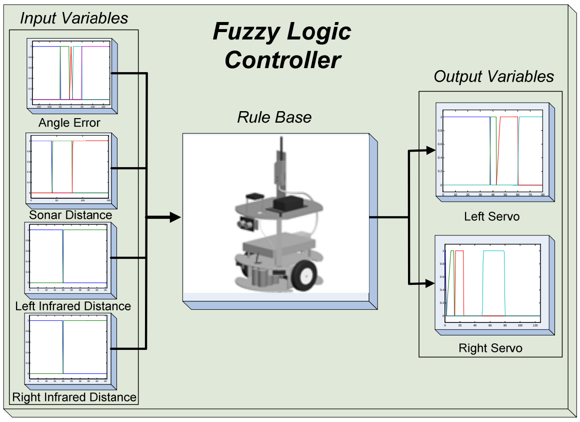

Fig. 2. The Fuzzy Logic Controller architecture of the HELOT robotic vehicle.

The input variables to the motion control block of Fig. 1: a) the four *collision possibilities* b) the *position_error* with linguistic values {*Backwards_1, Hard_Left, Left, Left2, Left1 Ahead, Right1, Right2, Right, Hard_Right, Backwards_2.*} The *position_error* takes is the difference between the desired and the actual heading of the vehicle. The output variables are: a) *translational_velocity* with linguistic variables {*back_full, back_normal, back_slow, stop, front_slow, front_normal, front_full*} b) *rotational_velocity* with linguistic variables {*right_full, right, right1, no_rotation, left1, left, left_full*}. The number of linguistic values for the position error, translational and rotational velocities is chosen after conducting several experiments to ensure smooth and accurate collision free navigation. If the value of the translational velocity is positive the vehicle moves forward; if it is negative the vehicle moves backwards. A positive rotational velocity results in vehicle turn left; a negative value in vehicle turn right. Navigation and collision avoidance are performed using rules of the type:

IF c_j is $LC^{(k)}$ AND θ is $L\Theta^{(k)}$ THEN tv is $LTV^{(k)}$ AND rv is $LTV^{(k)}$,

where k is the rule number, c_j is collision of type j, i.e., the output of the obstacle detection module, θ is the angle error, tv is the translational velocity and rv is the rotational velocity. $LC^{(k)}$, $L\Theta^{(k)}$, $LTV^{(k)}$, $RTV^{(k)}$ are the linguistic variables of c_j ,θ, tv, rv respectively. AND = min in all rules. The motion contol module of the suggested architecture applied to custom made robotic vehicles is presented in Fig. 2.

3.2 Unmanned Aerial Vehicles.

The starting point in dealing with *Unmanned Aerial Vehicles* (UAVs) has been the development of a simulation environment implemented in MATLAB with the fixed wing vehicle's motion dynamics adopted from the Aerosim Block Set integrated with SIMULINK. Implementations of the proposed navigation scheme for fixed wing unmanned aircrafts are presented in (Doitsidis et al., 2004) and in a similar approach in (Nikolos et al., 2003). Extensive real time experimentation has been done for the flight data of the UAV *Nearchos*, manufactured by EADS 3 SIGMA S.A. The UAV and real flight position data are shown in Fig. 3.

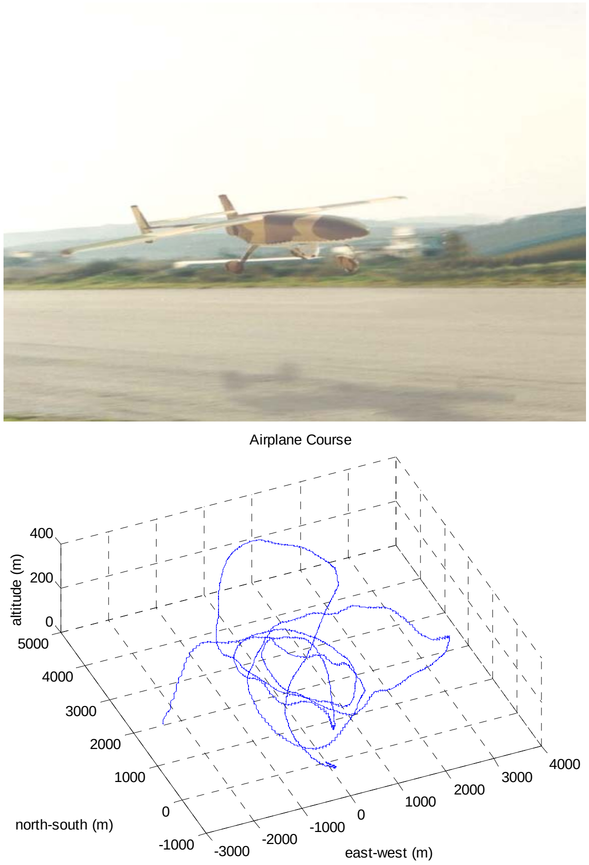

Fig. 3. The UAV *Nearchos* (property of EADS 3 SIGMA S. A.) and 3D position readings during a test flight.

In Doitsidis *et al.* (2004), a two module fuzzy logic based autonomous navigation system which is capable i) to fly through specified waypoints in a 3-D environment repeatedly, ii) to perform trajectory tracking, and, iii) to duplicate / follow another aerial vehicle, has been presented. The control modules are responsible for altitude and latitude-longitude control; when combined, they may adequately navigate the aerial vehicle. All input and output linguistic variables have a finite number of linguistic values with membership functions empirically defined. The altitude fuzzy logic controller has three inputs, that is a) altitude error, b) change of altitude error, and, c) airspeed. The altitude error is the difference between the desired altitude and the current altitude of the airplane. The change of altitude error indicates whether the aerial vehicle is approaching the desired altitude or if it is going away from it. The airspeed is the current speed of the vehicle. Outputs are the elevators command and the throttle command, responsible for the decent and accent of the aerial vehicle. The latitude-longitude controller has as inputs the heading error and the change of heading error. The heading error is the difference between the desired and the actual heading of the airplane. The output is the roll angle of the airplane.

A fuzzy controller for the autonomous navigation on the horizontal plane has been developed and presented in Nikolos *et al*. (2003). The controller inputs are the heading error of the aircraft and its current roll angle, while the output is the change command of the roll angle. The basic purpose of the navigation system was, to make the vehicle able to follow a predefined trajectory. The linguistic variables that represent the current roll angle are: *Right_Big (rb), Right_Medium (rm), Right_Small (rs), Zero, Left_Big (lb), Left_Medium (lm), Left_ Small (ls).* The second input to the fuzzy controller is the heading error, which is defined as the difference between the desirable and the factual direction of the aircraft. The factual direction is the heading of the aircraft, which is provided from the GPS. The desirable direction is the heading of a vector, with a starting point the current aircraft's position and ending point the desirable position. The linguistic variables that represent the heading error are: *Negative_Big (nb), Negative_Medium (nm), Negative_Small (ns), Zero, Positive_Big (pb), Positive_Medium (pm), Positive_Small (ps).* The desired and the actual heading direction take values ranging from 0^0 to 360^0, whereas the heading error takes values ranging from -180^0 to 180^0. However, in this implementation the heading error takes values in the region [-100^0, 100^0]. Negative (positive) values of heading error correspond to desirable right (left) roll. The linguistic variables that represent the heading error are: *Negative_Big (nb), Negative_Medium (nm), Negative_Small (ns), Zero, Positive_Big (pb), Positive_Medium (pm), Positive_Small (ps).* Experimental results of the fuzzy logic controller in terms of trajectory following are presented in Fig. 4. In this figure the continuous and discontinuous lines represent the desired and the actual trajectory respectively, the fuzzy logic controller forced the vehicle to follow.

Recent experimentation (Vitzilaios and Tsourveloudis, 2007) with Vertical Take Off and Landing (VTOL) UAVs showed that empirically tuned fuzzy controllers may be used for indoor helicopter hovering and way point navigation without prior knowledge of the system dynamics. The developed experimental test bed is shown in Fig. 5.

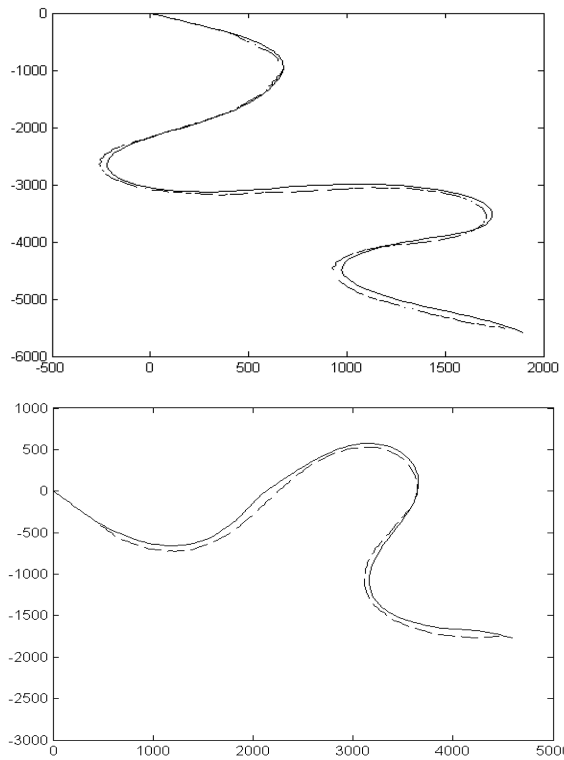

Fig. 4. Dashed lines represent the observed path of a UAV while solid lines are the desired trajectories.

Fig. 5. The experimental test bed for autonomous helicopter flight developed in the *Intelligent Systems and Robotics Laboratory* of the Technical University of Crete.

3.3 Autonomous Underwater Vehicles.

Most of the difficulties in navigation of *Autonomous Underwater Vehicles* (AUVs) are due to the inherently uncertain nature of these environments. In this section, we present an overview of the fuzzy logic implementations for the navigation of AUVs introduced in Tsourveloudis *et al.* (1998), and later developed in Kanakakis *et al.* (2004), and in Kanakakis and Tsourveloudis (2007).

Fuzzy logic navigation solutions have shown a good degree of robustness, which is crucial in the area of underwater robotics, where: 1) sonar data is unreliable, 2) mathematical models about the environment and the vehicles are usually not available, and 3) the only available navigation expertise is due to vehicle operators. The aim of underwater navigation is to guide the vehicle at a predefined point, by controlling various parameters such as pitch, yaw etc. The desired values of these parameters are the control objectives at any time instant. The fuzzy rules, which contain the navigation parameters, aim towards two goals: ideal trajectory tracking and collision avoidance.

Similar to the generic architecture described in Section 2, the adopted autonomous underwater scheme has the following modules:
A) The **sensor fusion/collision avoidance module**, where the readings of the sensors of the vehicle are provided to estimate the position of the vehicle and the collision possibility in all surrounding directions. The sensor fusion module is responsible for position monitoring and obstacle detection. As AUVs operate in unknown or poorly mapped ocean environments, static or moving obstacles find themselves in the desired path of the vehicle. In these cases the vehicle should be able to use it's on board sensors to monitor its position and to detect moving or static obstacles. This implies the use of a number of different kinds of sensors, like vision cameras, laser sensors, magnetic compasses, gyroscopic mechanisms and sonar sensors. For most cases where vision is poor, sonar sensors are used to estimate an underwater environment.
B) The **motion control module**, which performs low-level control of the vehicle's propellers, thrusters and fins in order to reach the determined goal point having the target surge velocity. The inputs are the goal point and the actual position and orientation, in earth-fixed coordinates, the target surge velocity and the vector of the actual vehicle velocities in body-fixed coordinates, and the sea current velocity.
Since the design of fuzzy controllers does mot require any strict modeling of the vehicle's behavior the above design is adopted for its simplicity, considering that it can be applied in all types of AUVs. Extensive simulation results using the mathematical model of the *Phoenix* AUV, (property of the Naval Postgraduate School at Monterey, California, USA), revealed that the adopted implementation of the control architecture may navigate the UAV without collisions under the presence of various sea currents. Many versions of underwater navigation fuzzy controllers have been tested within the suggested control architecture and comparisons of their performance are presented in Fig. 6.

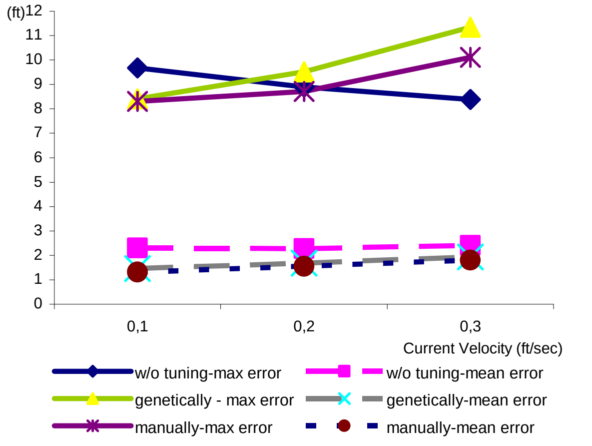

Fig. 6. Mean and Max error comparisons for three fuzzy controllers.

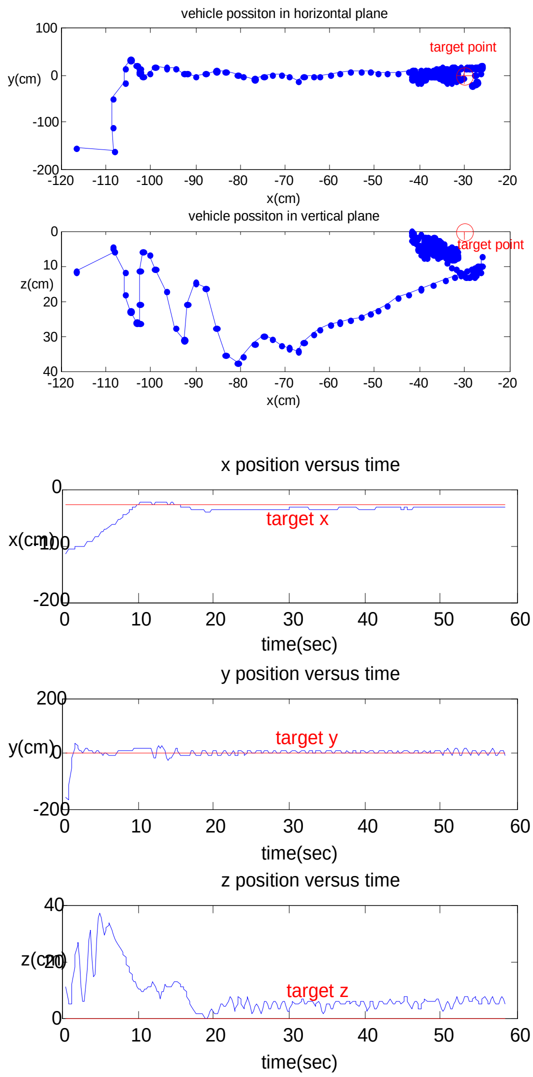

Fig. 7. Experimental results for the *Videoray* underwater vehicle.

Real time experiments with the Remotely Operated Vehicle (ROV) *Videoray* have verified the suggested approach. Fig. 7 presents monitoring of the ROV's autonomous motion towards a target position. The vehicle is controlled by a heuristic fuzzy controller.

4. CONCLUSION

This position paper presents a unified fuzzy control architecture which is developed and used by the *Intelligent Systems and Robotics Laboratory* of the Technical University of Crete, Greece, for the autonomous navigation of ground, aerial and underwater unmanned vehicles.

Fuzzy logic has been identified as a useful tool for developing controllers able to perform autonomous navigation. A two-layer control architecture has been described. The first layer is the sensor fusion module in which the vehicle evaluates readings from various sensors and interacts with the environment. In the second layer, the information derived from the previous layer is combined with other parameters i.e. heading, speed, altitude, position etc, and the actual commands that will move the vehicle towards its mission, are taken.

The diversity of presented results supports the claim that this architecture is effective regardless of vehicle type. Further research is needed towards the automatic adaptation of the suggested architecture to the navigation problem under scrutiny each time.

Acknowledgement: The author is grateful to his graduate students L. Doitsidis, E. Kanakakis, N. Vitzilaios, and to Prof. K. Valavanis, for their contributions.

REFERENCES

Doitsidis, L., K.P. Valavanis and N.C. Tsourveloudis (2002). Fuzzy Logic Based Autonomous Skid Steering Vehicle Navigation. *Proceedings of the IEEE International Conference on Robotics and Automation,* pp. 2171-2177, May 2002, Washington DC, U.S.A.

Doitsidis, L., K.P. Valavanis, N.C Tsourveloudis and M. Kontitsis (2004). A Framework for Fuzzy Logic Based UAV Navigation and Control. *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 4041-4046, April 2004, New Orleans, U.S.A.

Driankov, D. and A. Saffiotti (2001). *Fuzzy Logic Techniques for Autonomous Vehicle Navigation*. Physica – Verlag, Heidelberg.

Kanakakis, V. and N.C. Tsourveloudis (2007). Evolutionary Path Planning and Navigation of Autonomous Underwater Vehicles. *Proceedings of the 15th Mediterranean Conference on Control and Automation*, Athens, Greece, June 27-29.

Kanakakis, V., K.P. Valavanis and N.C. Tsourveloudis (2004). Fuzzy-Logic Based Navigation of Underwater Vehicles. *Journal of Intelligent and Robotic Systems*, **Vol. 40**, pp. 45-88.

Nikolos, I. K., L. Doitsidis, V.N. Christopoulos and N.C. Tsourveloudis (2003). Roll Control of Unmanned Aerial Vehicles using Fuzzy Logic. *WSEAS Transactions on Systems*, **Vol. 4**, pp. 1039-1047.

Piperidis S., L. Doitsidis, C. Anastasopoulos and N.C. Tsourveloudis (2007). A Low Cost Modular Robot Vehicle Design for Research and Education. *Proceedings of the 15th Mediterranean Conference on Control and Automation*, Athens, Greece, June 27-29.

Tsourveloudis N.C., L. Doitsidis, and K.P. Valavanis (2005). Autonomous Navigation of Unmanned Vehicles: A Fuzzy Logic Perspective. In: *Cutting Edge Robotics*, ARS Press, pp. 291-310.

Tsourveloudis, N.C., D. Gracanin, K.P. Valavanis. (1998). Design and testing of navigation algorithm for shallow water autonomous underwater vehicle. *Proceedings of the IEEE OCEANS 98*, Nice, France.

Tsourveloudis, N.C., K.P. Valavanis and T. Hebert (2001). Autonomous Vehicle Navigation Utilizing Electrostatic Potential Fields and Fuzzy Logic, *IEEE Transactions on Robotics and Automation*, **Vol. 17**, pp. 490-497.

Vitzilaios, N. and N. C. Tsourveloudis (2007). An Experimental Test Bed for the Flight Control of Unmanned Helicopters. Submitted to the 2008 *IEEE International Conference on Robotics and Automation*.